In the Claims

This listing of claims will replace all prior versions and listings of claims in this application.

1 (currently amended). A computer-implemented method for improving in dynamic routing tables processes for matching a highest-priority range with a destination address in $O(\log^2 n)$ time and inserting or deleting rules in $O(\log n)$ time comprising:

(a)——applying a top level balanced binary search tree (PTST) with at least one node comprising a lower level range search tree (RST) to a nonintersecting highest priority rule table comprising at least one nonintersecting range and corresponding priority, wherein the PTST comprises at most $2n$ nodes, with each of the PTST nodes associated with a point value;

(b)——applying a range allocation rule to allocate to each PTST node a subset of the nonintersecting ranges and corresponding priorities;

(c)——applying an RST to organize the subset of nonintersecting ranges and corresponding priorities allocated to each PTST node; and

(d)——maintaining balanced PTST, which can have up to $n$ empty nodes.

2 (previously presented). The method according to claim 1, wherein the PTST and RST are red-black trees.

3 (currently amended). The method according to claim 1, for matching a highest-priority range with a destination address, further comprising the steps of:

(a)——conducting a search of the PTST based on the destination address, wherein the search is conducted along a search path from a root of the PTST to a leaf of the PTST; and

(b)——examining the RST(s) of a PTST node along the search path for a best matching range and corresponding priority.

4 (currently amended). The method according to claim 1, for inserting a new rule, further comprising the steps of:

(a)——setting an initial node to a root node of the PTST in preparation of inserting a new range $r$;

(b)——conducting a search on the PTST for a node $z$ such that the range $r$ contains the point value of $z$;

(c)——if said node $z$ exists, inserting range $r$ into the RST($z$); and

(d)——if said node z does not exist, creating a new PTST node, inserting the new PTST node into the PTST, and inserting range $r$ into the RST of the new PTST node.

5 (original).  The method according to claim 4, wherein when the new PTST node is inserted into the PTST, the PTST is rebalanced.

6 (currently amended).  The method according to claim 1, for deleting a rule, further comprising the steps of:

(a)——conducting a search of the PTST for a node $z$ such that the range $r$ contains point($z$);

(b)——if node $z$ exists, deleting range $r$ from the RST of the node $z$;

(c)——if the RST of node $z$ becomes empty as a result of deleting range $r$ and node $z$ is a degree 0/1 node; deleting node $z$ from the PTST and rebalancing the PTST; and

(d)——when a size constraint has been violated, deleting a degree 0/1 PTST node having an empty RST and rebalancing the PTST.

7 (currently amended).  A computer-implemented method for improving in dynamic routing tables processes of matching a highest-priority prefix with a destination address and inserting or deleting new rules in $O(W)$ time comprising:

(a)——applying a top level balanced binary search tree (PTST) with at least one node comprising an array linear list (ALL) to a highest priority prefix table comprising at least one pair, wherein the pair comprises a prefix and corresponding priority, wherein the PTST comprises at most $2n$ nodes, with each of the PTST nodes associated with a point value;

(b)——applying a range allocation rule to each PTST node to allocate to each PTST node a subset of the pairs;

(c)——applying an ALL to organize the pairs allocated to each PTST node, wherein the ALL comprises a pair of prefix and corresponding priority; and

(d)——maintaining balanced PTST, ~~which can have~~ the balanced PTST having up to *n* empty nodes.

8 (previously presented). The method according to claim 7, wherein the PTST is a red-black tree.

9 (currently amended). The method according to claim 7, for matching a highest-priority prefix with a given destination address, further comprising the steps of:

(a)——conducting a search of the PTST based on the destination address, wherein the search is conducted along a search path from a root of the PTST to a leaf of the PTST; and

(b)——examining the ALL(s) of the PTST nodes along the search path for a best matching prefix and corresponding priority.

10 (currently amended). The method according to claim 7, for inserting a new rule, further comprising the steps of:

(a)——setting an initial node to a root node of the PTST in preparation of inserting a new prefix $p$;

(b)——conducting a search on the PTST for a node $z$ such that the prefix $p$ contains the point value of $z$;

(c)——if said node $z$ exists; inserting prefix $p$ into ALL($z$); and

(d)——if said node $z$ does not exist, creating a new PTST node, inserting the new PTST node into the PTST, and inserting prefix $p$ into an ALL of the new PTST node.

11 (original). The method according to claim 10, wherein when the new PTST node is inserted into the PTST, the PTST is rebalanced.

12 (currently amended). The method according to claim 7, for deleting a rule, further comprising the steps of:

(a) conducting a search of the PTST for a node $z$ such that the prefix $p$ contains point($z$);

(b) if such node $z$ exists, deleting prefix $p$ from the ALL of the node $z$;

(c) if the ALL becomes empty as a result of deleting prefix $p$ and node $z$ is a degree 0/1 node, deleting node $z$ from the PTST and rebalancing the PTST, and

(d) when a size constraint has been violated, deleting a degree 0/1 PTST node with an empty ALL and rebalancing the PTST.


13 (currently amended). A computer-implemented method for improving in dynamic routing tables processes for longest-prefix matching in $O(W)$ time and rule insertion and deletion in $O(\log n)$ time, comprising:

(a) applying a top level balanced binary search tree (PTST) with at least one node comprising a $W$-bit vector (bit($z$)) to a longest-matching prefix-table comprising at least one prefix, wherein the PTST comprises at most $2n$ nodes, with each of the PTST nodes associated with a point value;

(b) applying a range allocation rule to allocate to each PTST node a subset of prefixes;

(c) applying the bit($z$) to organize the prefixes allocated to each PTST node, wherein an $i$th position of bit($z$) is set to 1 if the prefix with length $i$ is allocated to a PTST node; and

(d) maintaining balance in the PTST.


14 (previously presented). The method according to claim 13, wherein the PTST is a red-black tree.


15 (currently amended). The method according to claim 13, for longest-prefix matching, further comprising the steps of:

(a) conducting a search of the PTST based on a given destination address, wherein the search is conducted along a search path from a root of the PTST to a leaf of the PTST; and

(b)——examining the bit($z$) vectors of PTST nodes along the search path to find a prefix with the longest matching length.

16 (currently amended). The method according to claim 13, for inserting a new rule, further comprising the steps of:

(a)——setting an initial node to a root node of the PTST in preparation of inserting a new prefix $p$;

(b)——conducting a search on the PTST nodes for the bit($z$) such that the prefix $p$ contains the point value of $z$;

(c)——if there exists a PTST node having the bit($z$) such that the prefix $p$ contains the point value of $z$, inserting prefix $p$ into the bit($z$); and

(d)——if there does not exist a PTST node having the bit($z$) such that the prefix $p$ contains the point value of $z$, creating a new PTST node, inserting the new PTST node into the PTST, and inserting prefix $p$ into the bit($z$) of the newly created PTST node.

17 (original). The method according to claim 16, wherein when the new PTST node is inserted into the PTST, the PTST is rebalanced.

18 (currently amended). The method according to claim 13, for deleting a rule, further comprising the steps of:

(a)——conducting a search of the PTST for a node $z$ such that the prefix $p$ contains point($z$);

(b)——if node $z$ is present, deleting prefix $p$ from the bit($z$);

(c)——if the bit($z$) vector becomes empty as a result of deleting prefix $p$ and node $z$ is a degree 0/1 node, deleting node $z$ from the PTST and rebalancing the PTST; and

(d)——when a size constraint has been violated, deleting a degree 0/1 PTST node having an empty bit($z$) and rebalancing the PTST.

19 (currently amended). A computer program product for improving in dynamic routing tables processes for matching a highest-priority range with a destination address and inserting or

deleting rules, the computer program product recorded on computer readable medium for routing packets comprising: a computer readable medium for receiving packets, a computer readable medium for identifying each rule matching the value of the packet; a computer readable medium for matching, inserting, or deleting rules in a dynamic, nonintersecting highest priority rule table (NHRT), and a computer readable medium for performing steps of manipulating the dynamic table, said steps comprising:

(a)——applying a top level balanced binary search tree (PTST) with at least one node comprising a lower level range search tree (RST) to a nonintersecting highest priority rule table comprising at least one nonintersecting range and corresponding priority, wherein the PTST comprises at most $2n$ nodes, with each of the PTST nodes associated with a point value;

(b)——applying a range allocation rule to allocate to each PTST node a subset of the nonintersecting ranges and corresponding priorities;

(c)——applying an RST to organize the subset of nonintersecting ranges and corresponding priorities allocated to each PTST node; and

(d)——maintaining balance in the PTST.


20 (previously presented). The computer program product according to claim 19, wherein the PTST and RST are red-black trees.


21 (currently amended). A computer program product for improving in dynamic routing tables processes of matching a highest-priority prefix with a destination address and inserting or deleting new rules, the computer program product recorded on computer readable medium for routing packets comprising: a computer readable medium for receiving packets, a computer readable medium for identifying each rule matching the value of the packet; a computer readable medium for matching, inserting, or deleting prefixes in a dynamic, highest priority prefix table (HPPT), and a computer readable medium for performing steps of manipulating the dynamic table, said steps comprising:

(a)——applying a top level balanced binary search tree (PTST) with at least one node comprising an array linear list (ALL) to a highest priority prefix table comprising at least one pair,

wherein the pair comprises a prefix and corresponding priority, wherein the PTST comprises at most $2n$ nodes, with each of the PTST nodes associated with a point value;

(b)——applying a range allocation rule to each PTST node to allocate to each PTST node a subset of the pair of prefix and corresponding priority;

(c)——applying an ALL to organize the prefixes and corresponding priorities allocated to each PTST node, wherein the ALL comprises the pair of prefix length and corresponding prefix priority; and

(e)——maintaining balance in the PTST.

22 (previously presented). The computer program product according to claim 21, wherein the PTST is a red-black tree.

23 (currently amended). A computer program product <u>for improving in dynamic routing tables processes for longest-prefix matching and rule insertion and deletion, the computer program product</u> recorded on computer readable medium for routing packets comprising: a computer readable medium for receiving packets, a computer readable medium for identifying each rule matching the value of the packet; a computer readable medium for matching, inserting, or deleting prefixes in a dynamic, longest-matching prefix table (LMPT), and a computer readable medium for performing steps of manipulating the dynamic table, said steps comprising:

(a)——applying a top level balanced binary search tree (PTST) with at least one node comprising a $W$-bit vector (bit($z$)) to a longest-matching prefix-table comprising at least one prefix, wherein the PTST comprises at most $2n$ nodes, with each of the PTST nodes associated with a point value;

(b)——applying a range allocation rule to allocate to each PTST node a subset of the prefixes;

(c)——applying the bit($z$) to organize the prefixes allocated to each PTST node, wherein an $i$th position of bit($z$) is set to 1 if the prefix with length $i$ is allocated to a PTST node; and

(d)——maintaining balance in the PTST.

24 (previously presented).  The computer program product according to claim 23, wherein the PTST is a red-black tree.